



## PYTHON H2 YETENEK SINAVI

### Hedefler

Bölüm 1: Mikro uygulamalar ile oyun için dökümantasyonu

Bölüm 2: Kodlama anı

Bölüm 3: Çalışma anı

### Senaryo

Eğlenceli bir oyun olan TicTacToe, namı diğer XoX oyununu yazacağız.

### Gereksinimler

- Mikro uygulamalar ile oyundaki ipuçları
- <https://playtictactoe.org/> ile oyun kuralları
- `python-h2.py` öğrenci script'i.

### Talimatlar

#### Bölüm 1: Mikro Uygulamalar İle Oyun İçin Dökümantasyon

Bu bölümde tictactoe oyununun bir bütün halinde düşünürsek eğer, içinde geçecek olan ipuçları çözeceğiz. Bu mikro uygulamalar size oyunu yazmanızda çok yardımcı olacak.

#### Adım 1: Board'u dizayn edelim

```
def DisplayBoard(board):
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")
    for row in range(3):
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
        for col in range(3):
            print("<!!!buradaki kodu değiştir>" + str(board[row][col]) + " ", end="")
        print("<!!!buradaki kodu değiştir>")
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")

board= [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
DisplayBoard(board)
```

```
PS C:\ python python-h2.py
```

```
+-----+-----+-----+
|       |       |       |
|   1   |   2   |   3   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   4   |   5   |   6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   7   |   8   |   9   |
|       |       |       |
+-----+-----+-----+
```

### Adım 2: Inline iç-içe For çok boyutlu liste oluşturalım

```
board = [[<!!!buradaki kodu değiştir> for j in range(3)] for i in range(3)]
print(board)
board[?][?] = 'X' # ortadaki kareyi 'X' olarak set et
print(board)
```

```
PS C:\ python python-h2.py
```

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 3], [4, 'X', 6], [7, 8, 9]]
```

### Adım 3: Digit değerlerin matris'e dönüştürülmesi

```
move = int(input("Lütfen Kare Seçin [1-9]: "))
move = <!!!buradaki kodu değiştir> # üç satır önemli. digit değerine sahip kareyi
row = <!!!buradaki kodu değiştir> # [row][col] değerine dönüştürmeliyiz. Örn
col = <!!!buradaki kodu değiştir> # 1 → [0][0], 2 → [0][1], 3 → [0][2] gibi
print(f"{row} {col}")
```

```
PS C:\> python python-h2.py
Lütfen Kare Seçin [1-9]: 2
0 1
PS C:\> py .\VSCoDe-Ecodation-PC\test1.py
Lütfen Kare Seçin [1-9]: 9
2 2
PS C:\> py .\VSCoDe-Ecodation-PC\test1.py
Lütfen Kare Seçin [1-9]: 7
2 0
PS C:\> py .\VSCoDe-Ecodation-PC\test1.py
Lütfen Kare Seçin [1-9]: 4
1 0
```

### Adım 4: Oyuncu-PC hamlelerini sırayla belirler

```
humanturn = True
while True:
    if humanturn:
        input("Sıra Kullanıcıda")
    else:
        input("Sıra PC'de")
    humanturn = <!!!buradaki kodu değiştir>
```

```
PS C:\> python python-h2.py
Sıra Kullanıcıda [enter]
Sıra PC'de [enter]
Sıra Kullanıcıda [enter]
Sıra PC'de [enter]
Sıra Kullanıcıda [enter]
Ctrl + C
```

### Adım 5: Satırlar-Sütunlar-Köşegenler kontrol edilir.

```
board= [[1, 2, 3], ['X', 'X', 'X'], [7, 8, 9]]
sgn = "X"
for rc in range(3):
    if <!!!buradaki kodu değiştir> == sgn: # satırları kontrol edelim
        print("X Kazandı")
```

```
PS C:\ python python-h2.py
```

```
X Kazandı
```

### Adım 6: Boş kareler kontrol edilir.

```
board= [[' ', ' ', ' '], ['X', 'O', 'X'], [' ', 'O', 'X']]
free = [] # boş liste tanımlanıyor
for row in range(3): # rows
    for col in range(3): # columns
        if board[row][col] not in ['O', 'X']: # boş kare mi?
            # evet ise append et
            free.append((row, col))
print(free)
```

```
PS C:\ python python-h2.py
```

```
[(0, 0), (0, 1), (0, 2), (2, 0)]
```

## Bölüm 2: Kodlama anı

**Not:** python-h2.py script'i kurumdan talep edin.

**Tavsiye:** Belki doğru rotada olacağınız ihtimalinden dolayı orjinal dosyayı Save As diyerek saklı kalmasını sağlayın. Baştan başlamak isteyebilirsiniz.

**Adım1: Randrange fonksiyonu için kütüphaneleri import edelim.**

```
<!!!buradaki kodu değiştir>
```

**Adım 2: Oyunun Başlangıcında ve Oyunun Herhangi Bir Anında Board'u Görüntüleyelim.**

```
def DisplayBoard(board):
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")
    for row in range(3):
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
        for col in range(3):
            print("<!!!buradaki kodu değiştir>" + str(board[row][col]) + " ", end="")
        print("<!!!buradaki kodu değiştir>")
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")
```

**Adım 3: Oyuncu Kare Seçiyor → O**

```
def EnterMove(board):
    ok = False
    while not ok:
        move = int(input("Lütfen Kare Seçin [1-9]: "))
        ok = <!!!buradaki kodu değiştir> # kullanıcı girişi doğru aralıkta mı?
        if not ok:
            print("[1-9] Arası Kare Seçmediniz. Lütfen Tekrar Giriniz!")
            continue
        move = <!!!buradaki kodu değiştir> # bu üç satır çok önemli. seçili digit değe
rine sahip kareyi
        row = <!!!buradaki kodu değiştir> # [row][col] değerine dönüştürmeliyiz. Örn
        col = <!!!buradaki kodu değiştir> # 1 → [0][0], 2 → [0][1], 3 → [0][2], 4
→ [1][0], 6 → [1][2] gibi
        sign = board[?][?] # seçili kareyi oku
        ok = sign not in ['O', 'X']
        if not ok: # seçili kare dolu ise tekrar denenmeli
            print("Seçili Kare. Lütfen Tekrar Giriniz!")
            continue
        board[?][?]= <!!!buradaki kodu değiştir> # seçili kareyi 'O' olarak set et
```

**Adım 4: Seçili Olmayan Kareleri Belirleme**

```
def MakeListOfFreeFields(board):
    free = [] # boş liste tanımlanıyor
    for row in range(3): # rows
        for col in range(3): # columns
            if <!!!buradaki kodu değiştir> not in ['O', 'X']: # boş kare mi?
                # evet ise append et
            <!!!buradaki kodu değiştir>
    return free
```

**Adım 5: Kazananı Belirleyelim**

```
def VictoryFor(board, sgn):
    if sgn == "X": # X için mi board'u kontrol edecek?
        who = 'me' # PC
    elif sgn == "O": # O için mi board'u kontrol edecek?
        who = 'you' # Kullanıcı
    cross1 = cross2 = True # for diagonals
    for rc in range(3):
        if <!!!buradaki kodu değiştir>: # satırları kontrol edelim
            return who
        if <!!!buradaki kodu değiştir>: # sütunları kontrol edelim
            return who
        if <!!!buradaki kodu değiştir> != sgn: # 1. köşegeni kontrol edelim
            cross1 = False
        if <!!!buradaki kodu değiştir> != sgn: # 2. köşegeni kontrol edelim
            cross2 = False
    if cross1 or cross2:
        return who
    return None
```

**Adım 6: PC Kare Seçiyor → X**

```
def DrawMove(board):
    free = MakeListOfFreeFields(board) # boş olan karelerden bir liste tanımlayalım
    cnt = len(free)
    if cnt > 0: # list boş değil ise, 'X' i set edeceğimiz kareyi random seçelim
        this = randrange(cnt)
        row, col = <!!!buradaki kodu değiştir>
        board[?][?] = <!!!buradaki kodu değiştir>
```

**Adım 7: Main blok**

```
board = [[<!!!buradaki kodu değiştir> for j in range(3)] for i in range(3)]
#print(board) #[[1, 2, 3], [4, 5 , 6], [7, 8, 9]]
board[?][?] = 'X' # ortadaki kareyi 'X' olarak set et
#print(board) #[[1, 2, 3], [4, 'X', 6], [7, 8, 9]]
free = MakeListOfFreeFields(board)
humanturn = True # sıra kim de? önce oyuncu başlayacak
while len(free):
    DisplayBoard(board)
    if humanturn:
        EnterMove(board)
        victor = VictoryFor(board, 'O')
    else:
        DrawMove(board)
        victor = VictoryFor(board, 'X')
    if victor != None:
        break
    humanturn = <!!!buradaki kodu değiştir>
    free = MakeListOfFreeFields(board)

DisplayBoard(board)
if victor == 'you':
    print("You won!")
elif victor == 'me':
    print("I won")
else:
    print("Tie!")
```

### Bölüm 3: Çalışma Anı

Powershell terminal ekranında:

PS C:\ `python python-h2.py`

```
+-----+-----+-----+
|       |       |       |
|   1   |   2   |   3   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   4   |   X   |   6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|   7   |   8   |   9   |
|       |       |       |
+-----+-----+-----+
```

Lütfen Kare Seçin [1-9]: 0

[1-9] Arası Kare Seçmediniz. Lütfen Tekrar Giriniz!

Lütfen Kare Seçin [1-9]: 10

[1-9] Arası Kare Seçmediniz. Lütfen Tekrar Giriniz!



+-----+-----+-----+						
	1		2		3	
+-----+-----+-----+						
	4		X		6	
+-----+-----+-----+						
	7		8		9	
+-----+-----+-----+						

Lütfen Kare Seçin [1-9]: 1

+-----+-----+-----+						
	0		2		3	
+-----+-----+-----+						
	4		X		6	
+-----+-----+-----+						
	X		8		9	
+-----+-----+-----+						

## PYTHON H2 YETENEK SINAVI

---

```
+-----+-----+-----+
|       |       |       |
|  O    |    2   |    3   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  4    |    X   |    6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  X    |    8   |    9   |
|       |       |       |
+-----+-----+-----+
```

Lütfen Kare Seçin [1-9]: 3

```
+-----+-----+-----+
|       |       |       |
|  O    |    2   |    O   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  4    |    X   |    6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  X    |    8   |    X   |
|       |       |       |
+-----+-----+-----+
```

Lütfen Kare Seçin [1-9]: 2

```
+-----+-----+-----+
|       |       |       |
|  O    |    O   |    O   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  4    |    X   |    6   |
|       |       |       |
+-----+-----+-----+
|       |       |       |
|  X    |    8   |    X   |
|       |       |       |
+-----+-----+-----+
```

You won!

PS C:\>

## Öğrenci Script'i

```
#####

# Bu program:
# - Inline iç-içe for ile [[1, 2, 3], [4, 'X', 6], [7, 8, 9]] çok boyutlu liste oluşturur
# - Randrange fonksiyonu için random modülü ekler
# - For döngüsü ile board'u ekrana basar
# - Digit olarak seçilen karelerin (1, 2, 3, 4, 6 ...)
# - Matrise dönüştülmesini sağlar 1 → [0][0], 2 → [0][1], 3 → [0][2], 4 → [1][0], 6 → [1][2] gibi
# - Belirli aralıkta değer girdirmeyi başarır
# - break-continue doğru anda kullanmayı bilir
#
# Öğrenci:
# 1. Randrange fonksiyonu için kütüphaneleri import edecek.
# 2. Oyunun Başlangıcında ve Oyunun Herhangi Bir Anında Board'u Görüntüleyecek
# 3. Oyuncu Kare Seçecek → 0
# 4. Seçili Olmayan Kareleri Belirleyecek
# 5. Kazananı Belirleyecek
# 6. PC Kare Seçecek → X
# 7. Main Block Tasarlayacak
#####

# 1 → Randrange fonksiyonu için kütüphaneleri import edelim.
<!!!buradaki kodu değiştir>

# 2 → Oyunun Başlangıcında ve Oyunun Herhangi Bir Anında Board'u Görüntüleyelim
def DisplayBoard(board):
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")
    for row in range(3):
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
        for col in range(3):
            print("<!!!buradaki kodu değiştir>" + str(board[row][col]) + " ", end="")
        print("<!!!buradaki kodu değiştir>")
        print("<!!!buradaki kodu değiştir>" * 3, "|", sep="")
    print("<!!!buradaki kodu değiştir>" * 3, "+", sep="")

# 3 → Oyuncu Kare Seçiyor → 0
def EnterMove(board):
    ok = False
    while not ok:
```

## PYTHON H2 YETENEK SINAVI

---

```
move = int(input("Lütfen Kare Seçin [1-9]: "))
ok = <!!!buradaki kodu değiştir> # kullanıcı girişi doğru aralıkta mı?
if not ok:
    print("[1-9] Arası Kare Seçmediniz. Lütfen Tekrar Giriniz!")
    continue

move = <!!!buradaki kodu değiştir> # bu üç satır çok önemli. seçili digit
değerine sahip kareyi

row = <!!!buradaki kodu değiştir> # [row][col] değerine dönüştürmeliyiz. Örn
col = <!!!buradaki kodu değiştir> # 1 → [0][0], 2 → [0][1], 3 → [0][2], 4
→ [1][0], 6 → [1][2] gibi

sign = board[?][?] # seçili kareyi oku
ok = sign not in ['O', 'X']
if not ok: # seçili kare dolu ise tekrar denenmeli
    print("Seçili Kare. Lütfen Tekrar Giriniz!")
    continue

board[?][?] = <!!!buradaki kodu değiştir> # seçili kareyi 'O' olarak set et

# 4 → Seçili Olmayan Kareleri Belirleme
def MakeListOfFreeFields(board):
    free = [] # boş liste tanımlanıyor
    for row in range(3): # rows
        for col in range(3): # columns
            if <!!!buradaki kodu değiştir> not in ['O', 'X']: # boş kare mi?
                # evet ise append et
                <!!!buradaki kodu değiştir>
    return free

# 5 → Kazananı Belirleyelim
def VictoryFor(board, sgn):
    if sgn == "X": # X için mi board'u kontrol edecek?
        who = 'me' # PC
    elif sgn == "O": # O için mi board'u kontrol edecek?
        who = 'you' # Kullanıcı
    cross1 = cross2 = True # for diagonals
    for rc in range(3):
        if <!!!buradaki kodu değiştir>: # satırları kontrol edelim
            return who
        if <!!!buradaki kodu değiştir>: # sütunları kontrol edelim
            return who
    if <!!!buradaki kodu değiştir> != sgn: # 1. köşegeni kontrol edelim
        cross1 = False
```

```
        if <!!!buradaki kodu değiştir> != sgn: # 2. köşegeni kontrol edelim
            cross2 = False
    if cross1 or cross2:
        return who
    return None

# 6 → PC Kare Seçiyor → X
def DrawMove(board):
    free = MakeListOfFreeFields(board) # boş olan karelerden bir liste tanımlayalım
    cnt = len(free)
    if cnt > 0: # list boş değil ise, 'X' i set edeceğimiz kareyi random seçelim
        this = randrange(cnt)
        row, col = <!!!buradaki kodu değiştir>
        board[?][?] = <!!!buradaki kodu değiştir>

# 7 → main
board = [[<!!!buradaki kodu değiştir> for j in range(3)] for i in range(3)]
#print(board) #[[1, 2, 3], [4, 5 , 6], [7, 8, 9]]
board[?][?] = 'X' # ortadaki kareyi 'X' olarak set et
#print(board) #[[1, 2, 3], [4, 'X', 6], [7, 8, 9]]
free = MakeListOfFreeFields(board)
humanturn = True # sıra kim de? önce oyuncu başlayacak
while len(free):
    DisplayBoard(board)
    if humanturn:
        EnterMove(board)
        victor = VictoryFor(board, 'O')
    else:
        DrawMove(board)
        victor = VictoryFor(board, 'X')
    if victor != None:
        break
    humanturn = <!!!buradaki kodu değiştir>
    free = MakeListOfFreeFields(board)

DisplayBoard(board)
if victor == 'you':
    print("You won!")
elif victor == 'me':
    print("I won")
```

## PYTHON H2 YETENEK SINAVI

---

```
else:  
    print("Tie!")
```

KAYNAKÇA/REFERANS

- 1→ <https://www.netacad.com/>
- 2→ <https://pythoninstitute.org/>